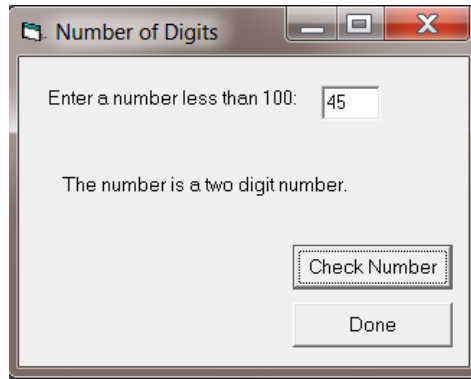


# Chapter 4

## Programs

---

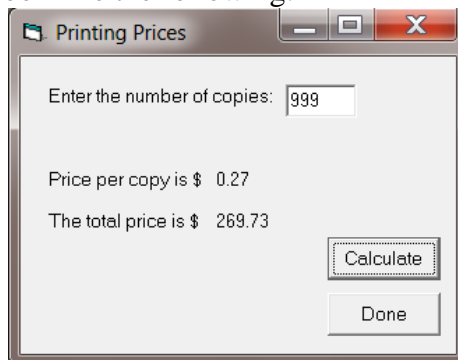
1. Create a Number of Digits application that allows the user to enter a number less than 100 and then displays a message stating whether the number is a one digit number or a two digit number. The application interface should look like the following.



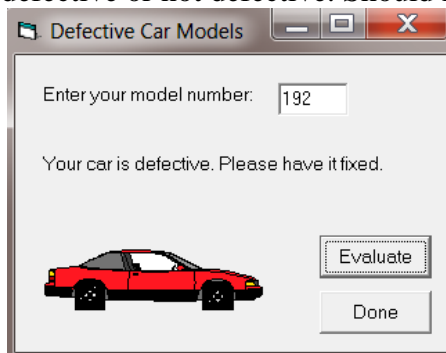
2. The Printing Place has different printing prices based on the number of copies to be printed:

0-499 copies	\$0.30 per copy
500-749 copies	\$0.28 per copy
750-999 copies	\$0.27 per copy
1000 copies or more	\$0.25 per copy

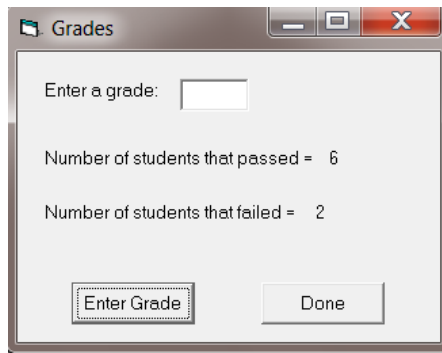
Create a Printing Prices application that ask the user for the number of copies and then calculates the total price. The program should look like the following:



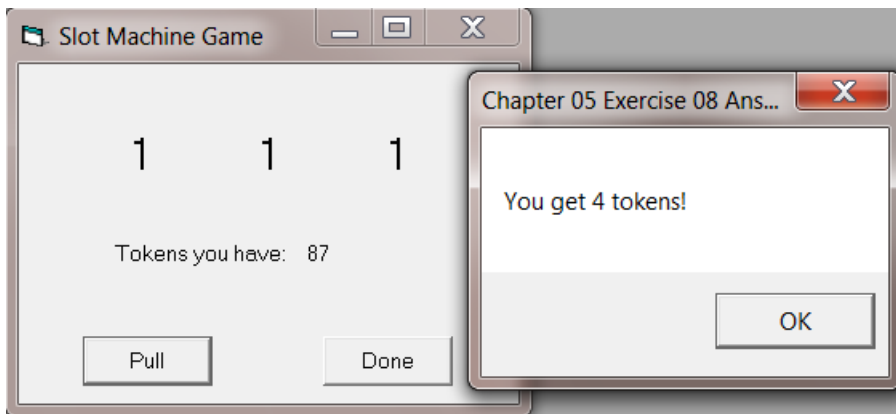
3. The Auto Company produced some models of cars that have defects. Cars with model numbers 119,179,189, through 195, 221, and 780 have been found to have the defect. Create a Defective Car program that allows the user to enter the model number to find out if it is defective. A car graphic should be display if the car is defective and not displayed if it is not defective. The program also displays if text stating if the car is defective or not defective. Should look like the program below.



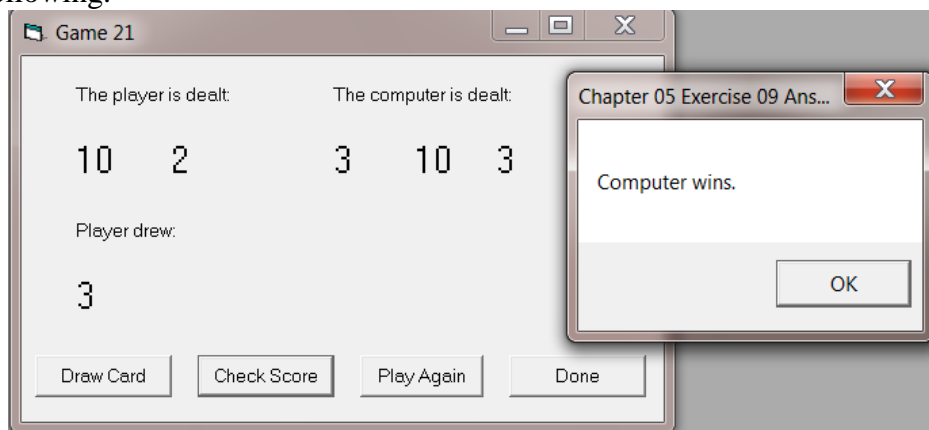
4. Create a Grades application that reads in letter grades and continuously displays the number of students who passed (D or better) and the number who failed. The program interface should look like the following.



5. Create a Slot Machine Program acts like a simple slot machine. The user starts with 100 tokens. With each "pull" of the handle, the user loses 1 token and the computer "spins" three wheels, each consisting of the numbers 1, 2 and 3. If all three numbers are 1, the user gets 4 tokens; if all are 2, the user gets 8 tokens; if all are 3, the user gets 12 tokens. The number of tokens that the user has should be displayed on the form and the results of the spin should be displayed in a message box. Should look similar to the following.



6. Create a Game 21 program to simulate the version of the game "21". A deck of cards numbered 1 through 10 is used and any number can be repeated. The computer starts by dealing you two randomly picked cards, and deals itself three randomly picked cards that are not revealed until Check Score button is clicked. You may then draw as many cards as you want, one by one. If both scores are over 21, or both scores are equal but under 21, the game is declared a draw. Otherwise, the winner is the one with the highest total and under or equal to 21. The result is displayed in a message box. Program should look similar to the following.



7. Create a Sandwich Order program that allows the user to generate a sandwich order that includes the size of the sandwich and the fixings for the sandwich. A small sandwich is \$2.50 and a large sandwich is \$4.00. Mustard and Mayonnaise are free, lettuce and onion are \$0.10 each, tomato is \$0.25, and cheese is \$0.50. The defaults should be a small sandwich with no fixings. Update the price when fixings are deselected or sandwich size changes. The program should calculate the total price of the order and print the order.

The screenshot shows a window titled "Sandwich Order" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a form with the following elements:

- Sandwich size:** A group box containing two radio buttons: "small" (selected) and "large".
- Fixings:** A section with six checkboxes arranged in two rows:
  - Row 1: ☐ Lettuce, ☒ Tomato, ☒ Mustard
  - Row 2: ☐ Onion, ☒ Cheese, ☐ Mayonnaise
- Total Price:** A label "Total Price = \$" followed by the value "3.25".
- Buttons:** Two buttons at the bottom right: "Print" and "Done".